

AMENDMENTS TO THE CLAIMS

Please add claims 38-48 as follows.

1 1. (Previously Presented) A method of searching for a string in a lexical cache,  
2 comprising the computer-implemented steps of:  
3 generating a key based on the string;  
4 selecting a lexical container from among a plurality of lexical containers based on a  
5 length of the key, said lexical containers associated with respective key  
6 lengths and configured to hold respective maximum numbers of entries based  
7 on the respective key lengths; and  
8 searching the selected lexical container for an entry associated with the string based  
9 on the key,  
10 wherein at least one of the lexical containers is configured to hold a different  
11 maximum number of entries than at least another one of the lexical containers.

1 2. (Original) The method of claim 1, wherein the step of generating a key based  
2 on the string includes the step of compressing the string to produce the key.

1 3. (Original) The method of claim 2, wherein the step of compressing the string  
2 to produce the key includes the step of performing an n-gram compression on the  
3 string.

1 4. (Original) The method of claim 1, wherein the step of generating a key based  
2 on the string includes the step of using the string as the key.

1 5. (Previously Presented) The method of claim 1, wherein the step of selecting a  
2 lexical container includes the steps of:  
3 generating a prefix based on the key; and  
4 selecting the lexical container from among the plurality of the lexical containers  
5 based on the length of the key and the prefix.

1 6. (Previously Presented) A method of searching for a string in a lexical cache,  
2 comprising the computer-implemented steps of:  
3 generating a key based on the string;  
4 identifying a hash table from among a plurality of hash tables based on the length of  
5 the key, said hash table containing sequences of slots for holding entries  
6 associated with strings, each of said sequences of slots corresponding to a  
7 respective hash value, wherein at least one of the hash tables is configured to  
8 hold a different number of slots than at least another one of the hash tables;  
9 computing a hash value based on the key; and  
10 searching the hash table based on the hash value for a slot holding an entry associated  
11 with said string.

1 7. (Original) The method of claim 6, wherein the step of computing a hash value  
2 based on the key includes the step of computing the hash value based on the key and  
3 a prime number associated with the hash table.

1     8.     (Original) The method of claim 7, wherein the step of searching the hash table  
2     based on the hash value includes the steps of:  
3     indexing one or more fixed regions of the hash table, each of the fixed regions having  
4     the prime number of slots, based on the hash value to identify one or more  
5     respective slots; and  
6     inspecting the one or more respective slots for a respective key value matching the  
7     key.

1     9.     (Original) The method of claim 8, wherein the step of searching the hash table  
2     further includes the step of searching for the key in a linked list of slots stored in an  
3     expansion region of the hash table, if the key was not found in the one or more  
4     respective slots for the key.

1     10.    (Original) The method of claim 6, further including the step of, if an entry for  
2     the string is not found at a first slot that corresponds to the hash value, but is found in  
3     a slot that belongs to a sequence of slots that correspond to keys that produce said  
4     hash value, then moving a relative position of the entry for the string within the  
5     sequence of slots toward the beginning of the sequence of slots.

1     11.    (Original) The method of claim 6, further comprising the step of initializing a  
2     descriptor for the hash table, said descriptor storing a reference to the hash table and  
3     parameters for the hash table;

4            wherein the step of identifying a hash table includes the step of identifying a  
5            descriptor indicating the hash table and a prime number.

1    12.    (Original) The method of claim 11, wherein the step of initializing a descriptor  
2    for the hash table includes the step of initializing a prime number for use in computing a  
3    hash value.

1    13.    (Original) The method of claim 11, wherein the step of initializing a descriptor  
2    for the hash table includes the step of initializing a maximum number of slots for the hash  
3    table.

1    14.    (Original) The method of claim 11, wherein the step of initializing a descriptor for  
2    the hash table includes the step of initializing a maximum length of the sequences of slots  
3    for the hash table.

1    15.    (Previously Presented) A method of searching for a string in a lexical cache,  
2    comprising the computer-implemented steps of:

3    compressing the string to generate a key;

4    identifying a hash table from among a plurality of hash tables based on a length of the

5    key, said hash table containing sequences of slots for holding respective key

6    values, each of said sequences of slots corresponding to a respective hash value

7    and a number of slots

8    being based on a respective key length, wherein at least one of the hash tables is

9    configured to hold a different number of slots than at least another one of the hash tables;

10     computing a hash value based on the key;  
11     using said hash value to locate a beginning of the particular sequence of slots that  
12         correspond to said hash value;  
13     searching the particular sequence of slots for a slot holding a key value matching the key;  
14     and if a slot having a key value matching the key is found in the particular sequence of  
15     slots, but is not at  
16         the beginning of said particular sequence of slots, then moving a relative position  
17     of the key value within the particular sequence of slots toward the beginning of the  
18     particular sequence of slots.

1     16.     (Previously Presented) A computer-readable storage medium bearing  
2     instructions for searching for a string in a lexical cache, said instructions arranged,  
3     when executed by one or more processors, to cause the one or more processors to  
4     perform the steps of:  
5         generating a key based on the string;  
6     selecting a lexical container from among a plurality of lexical containers based on a  
7         length of the key, said lexical containers associated with respective key  
8         lengths and configured to hold respective maximum numbers of entries based  
9         on the respective key lengths; and searching the selected lexical container for  
10     an entry associated with the string based on the key,

11 wherein at least one of the lexical containers is configured to hold a different  
12 maximum number of entries than at least another one of the lexical containers.

1 17. (Original) The computer-readable medium of claim 16, wherein the step of  
2 generating a key based on the string includes the step of compressing the string to  
3 produce the key.

1 18. (Original) The computer-readable medium of claim 17, wherein the step of  
2 compressing the string to produce the key includes the step of performing an n-gram  
3 compression on the string.

1 19. (Original) The computer-readable medium of claim 16, wherein the step of  
2 generating a key based on the string includes the step of using the string as the key.

1 20. (Previously Presented) The computer-readable medium of claim 16, wherein  
2 the step of selecting a lexical container includes the steps of:

3 generating a prefix based on the key; and

4 selecting the lexical container from among the plurality of the lexical containers

5 based on the length of the key and the prefix.

1 21. (Previously Presented) A computer-readable storage medium bearing  
2 instructions for searching for a string in a lexical cache, said instructions arranged,  
3 when executed by one or more processors, to cause the one or more processors to  
4 perform the steps of:

5 generating a key based on the string;

6 identifying a hash table from among a plurality of hash tables based on the length of  
7 the key, said hash table containing sequences of slots for holding entries  
8 associated with strings, each of said sequences of slots corresponding to a  
9 respective hash value, wherein at least one of the hash tables is configured to  
10 hold a different number of slots than at least another one of the hash tables;  
11 computing a hash value based on the key; and  
12 searching the hash table based on the hash value for a slot holding an entry associated  
13 with said string.

1 22. (Original) The computer-readable medium of claim 21, wherein the step of  
2 computing a hash value based on the key includes the step of computing the hash  
3 value based on the key and a prime number associated with the hash table.

1 23. (Original) The computer-readable medium of claim 22, wherein the step of  
2 searching the hash table based on the hash value includes the steps of:  
3 indexing one or more fixed regions of the hash table, each of the fixed regions having  
4 the prime number of slots, based on the hash value to identify one or more  
5 respective slots; and

6 inspecting the one or more respective slots for a respective key value matching the  
7 key.

1 24. (Original) The computer-readable medium of claim 23, wherein the step of  
2 searching the hash table further includes the step of searching for the key in a linked

3 list of slots stored in an expansion region of the hash table, if the key was not found  
4 in the one or more respective slots for the key.

1 25. (Original) The computer-readable medium of claim 21, wherein said  
2 instructions are further arranged to cause the one or more processors to perform the  
3 step of, if an entry for the string is not found at a first slot that corresponds to the  
4 hash value, but is found in a slot that belongs to a sequence of slots that correspond  
5 to keys that produce said hash value, then moving a relative position of the entry for  
6 the string within the sequence of slots toward the beginning of the sequence of slots.

1 26. (Original) The computer-readable medium of claim 21, wherein said  
2 instructions are further arranged to cause the one or more processors to perform the  
3 step of initializing a descriptor for the hash table, said descriptor storing a reference  
4 to the hash table and parameters for the hash table;  
5 wherein the step of identifying a hash table includes the step of identifying a  
6 descriptor indicating the hash table and a prime number.

1 27. (Original). The computer-readable medium of claim 26, wherein the step of  
2 initializing a descriptor for the hash table includes the step of initializing a prime  
3 number for use in computing a hash value.

1 28. (Original) The computer-readable medium of claim 26, wherein the step of  
2 initializing a descriptor for the hash table includes the step of initializing a maximum  
3 number of slots for the hash table.



1 29. (Original) The computer-readable medium of claim 26, wherein the step of  
2 initializing a descriptor for the hash table includes the step of initializing a maximum  
3 length of the sequences of slots for the hash table.

1 30. (Previously Presented) A computer-readable storage medium bearing  
2 instructions for searching for a string in a lexical cache, said instructions arranged,  
3 when executed by one or more processors, to cause the one or more processors to  
4 perform the steps of:

5 compressing the string to generate a key;

6 identifying a hash table from among a plurality of hash tables based on a length of the

7 key, said hash table containing sequences of slots for holding respective key

8 values, each of said sequences of slots corresponding to a respective hash

9 value and a number of slots being based on a respective key length, wherein at

10 least one of the hash tables is configured to hold a different number of slots

11 than at least another one of the hash tables;

12 computing a hash value based on the key;

13 using said hash value to locate a beginning of the particular sequence of slots that

14 correspond to said hash value;

15 searching the particular sequence of slots for a slot holding a key value matching the

16 key; and if a slot having a key value matching the key is found in the particular

17 sequence of slots, but is not at the beginning of said particular sequence of slots, then

18 moving a relative position of the key value within the particular sequence of slots  
19 toward the beginning of the particular sequence of slots.

1 31. (Previously Presented) A method of storing a string in a lexical cache, comprising  
2 the computer-implemented steps of:

3 generating a key based on the string;

4 selecting a lexical container from among a plurality of lexical containers based on a

5 length of the key, said lexical containers are associated with respective key

6 lengths and configured to hold respective maximum numbers of entries based

7 on the respective key lengths; and

8 storing the string in an entry in the selected lexical container based on the key,

9 wherein at least one of the lexical containers is configured to hold a different maximum

10 number of entries than at least another one of the lexical containers.

1 32. (Previously Presented) A computer-readable storage medium bearing

2 instructions for storing a string in a lexical cache, said instructions arranged, when

3 executed by one or more processors, to cause the one or more processors to perform the

4 steps of:

5 generating a key based on the string;

6 selecting a lexical container from among a plurality of lexical containers based on a

7 length of the key, wherein the lexical containers are associated with respective

8 key lengths and configured to hold respective maximum numbers of entries  
9 based on the respective key lengths; and  
10 storing the string in an entry in the selected lexical container based on the key,  
11 wherein at least one of the lexical containers is configured to hold a different  
12 maximum number of entries than at least another one of the lexical containers.

1 33. (Previously Presented) The method of claim 1, wherein:

2 a first lexical container of the lexical containers is associated with a first key length;  
3 a second lexical container of the lexical containers is associated with a second key  
4 length; the first key length is less than the second key length; and  
5 the first lexical container is configured to hold more entries than the second lexical  
6 container.

1 34. (Previously Presented) The method of claim 31, wherein:

2 a first lexical container of the lexical containers is associated with a first key length;  
3 a second lexical container of the lexical containers is associated with a second key  
4 length; the first key length is less than the second key length; and  
5 the first lexical container is configured to hold more entries than the second lexical  
6 container.

1 35. (Previously Presented) A method of providing a lexical cache, comprising the  
2 computer-implemented steps of:

3 allocating a plurality of lexical containers each configured to contain a respective  
4 maximum number of entries based on a function that includes a term that is  
5 inversely proportional to a logarithm of a key length associated with the lexical  
6 containers; and

7 searching for one of the entries associated with a string within one of the plurality of  
8 lexical containers corresponding to a key generated based on the string.

1 36. (Canceled)

2 37. (Previously Presented) The method of claim 1, wherein the step of searching  
3 the selected lexical container includes searching only the selected lexical container.

1 38. (New) A computer-readable medium carrying one or more sequences of  
2 instructions which, when executed by one or more processors, causes the one or more  
3 processors to perform the method recited in Claim 33.

1 39. (New) A computer-readable medium carrying one or more sequences of  
2 instructions which, when executed by one or more processors, causes the one or more  
3 processors to perform the method recited in Claim 34.

1 40. (New) A computer-readable medium carrying one or more sequences of  
2 instructions which, when executed by one or more processors, causes the one or more  
3 processors to perform the method recited in Claim 35.

4    41.    (New) A computer-readable medium carrying one or more sequences of  
5    instructions which, when executed by one or more processors, causes the one or more  
6    processors to perform the method recited in Claim 37.